Loss Functions Explained

Reference: https://www.youtube.com/watch?v=gb5nm_3jBIo

Loss in Machine learning helps us understand the difference between the predicted value & the actual value.

$$Error = Y - \hat{Y}$$

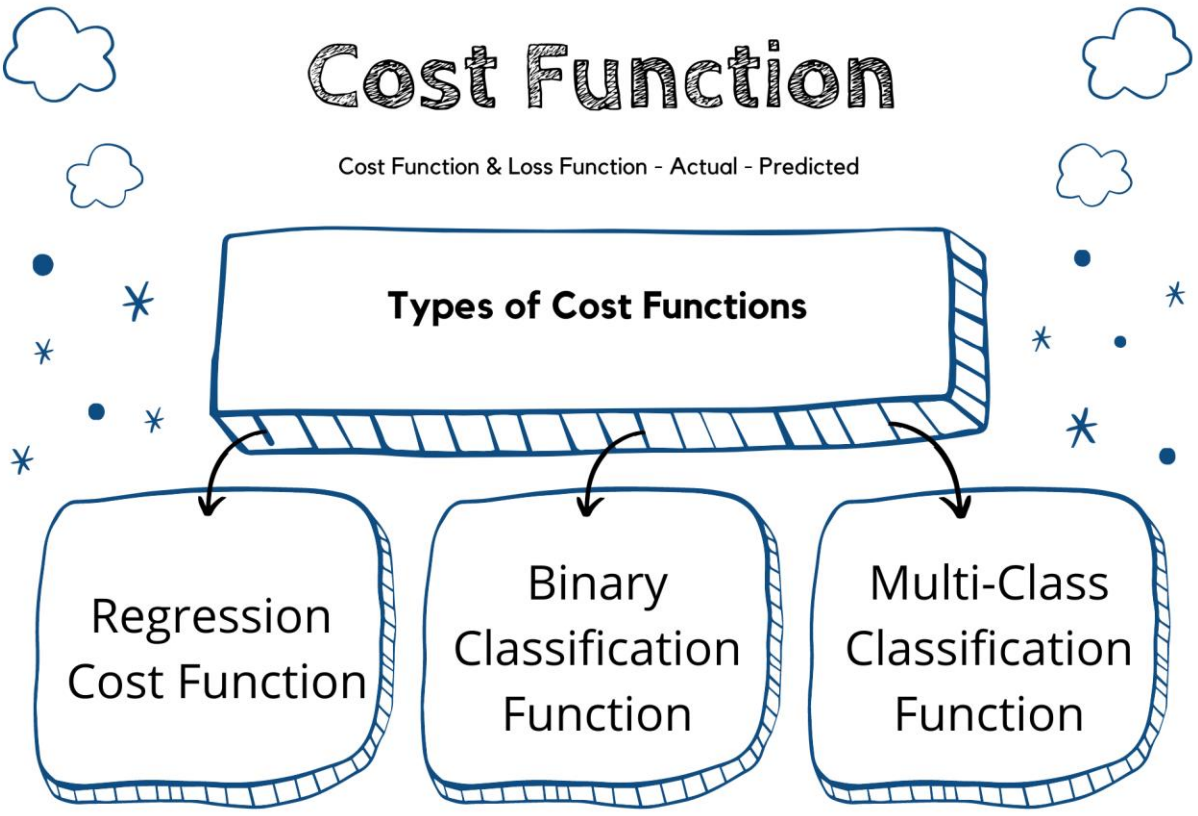
$$Y = Actual \quad \hat{Y} = Predicted$$

Difference between Loss function and Cost function:

Loss function: Used when we refer to the error for a single training example.

Cost function: Used to refer to an average of the loss functions over an entire training data.

Cost Function is generic term for model.



Regression cost Functions

A cost function used in the regression problem is called “Regression Cost Function”. They are calculated on the distance-based error.

The most used Regression cost functions are below:

1. Mean Error (ME):

- In this cost function, the error for each training data is calculated and then the mean value of all these errors is derived.
- Calculating the mean of the errors is the simplest and most intuitive way possible.
- The errors can be both negative and positive. So, they can cancel each other out during summation giving zero mean error for the model.
- Thus, this is not a recommended cost function, but it does lay the foundation for other cost functions of regression models.

2. Mean Squared Error (MSE) (aka L2 loss)

- This improves the drawback we encountered in Mean Error above. Here a square of the difference between the actual and predicted value is calculated to avoid any possibility of negative error.
- It is measured as the average of the sum of squared differences between predictions and actual observations.

$$\mathrm{MSE} = \frac{\sum_{i=0}^{n} (\mathrm{Y} - \hat{\mathrm{Y}})^2}{\mathrm{n}}$$

- It is also known as L2 loss, as it represents a (a - b) ^2 I.e. a^2 + 2ab + b^2 a quadratic equation and if we plot quadratic then it only gives global minima which is an advantage to get model optimize better but fails if data has outliers.
- In MSE, since each error is squared, it helps to penalize even small deviations in prediction when compared to MAE. But if our dataset has outliers that contribute to

larger prediction errors, then squaring this error further will magnify the error many times more and also lead to higher MSE error.

- Hence, we can say that it is less robust to outliers. (or sensitive to outliers)

3. Mean Absolute Error (MAE) (aka L1 loss)

- This cost function also addresses the shortcoming of mean error differently. Here an absolute difference between the actual and predicted value is calculated to avoid any possibility of negative error.
- So, in this cost function, MAE is measured as the average of the sum of absolute differences between predictions and actual observations.

$$\mathrm{MAE} = \frac{\sum_{\mathrm{i}\ =0}^{\mathrm{n}} |\mathrm{Y} - \hat{\mathrm{Y}}|}{\mathrm{n}}$$

- It is also known as L1 Loss, it may have local minima.
- It is robust to outliers thus it will give better results even when our dataset has noise or outliers.

4. Huber Loss (Best of both worlds I.e MAE and MSE)

- The Huber Loss offers the best of both worlds by balancing the MSE and MAE together. We can define it using the following piecewise function:

$$L_\delta(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{for} |y - f(x)| \leq \delta, \\ \delta\,|y - f(x)| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases}$$

- What this equation essentially says is: for loss values less than delta, use the MSE; for loss values greater than delta, use the MAE. This effectively combines the best of both worlds from the two loss functions!
- Where the delta value is hyperparameter to tune.

Classification cost Functions

---

1. **Binary Cross-Entropy (aka Log Loss)**

- It is used in binary classification problems like two classes.
- It measures the dissimilarity between predicted and target probability distributions.

$$\text{Log Loss} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i)$$

Here, y represents the true binary label (0 or 1), ŷ represents the predicted probability, and log represents the natural logarithm.

2. **Multiclass Cross-Entropy / Categorical Cross-Entropy**

- Categorical Cross entropy is used for Multiclass classification and SoftMax regression.
- It quantifies the difference between predicted and target probability distributions.

Note: Check for SoftMax function, one hot encoding, categorical cross-entropy, sparse categorical cross entropy.